

Change config of ChameleonEC

```
cd ~/ElasticRepair/conf
vim config.xml
cd ~/ElasticRepair/scripts
python3 edit_config.py
```

YCSB-Evaluation:

Start HBase:

```
start-all.sh
```

```
root@izbp1c010bxn27ezgr1qbZ:~# start-all.sh
WARNING: HADOOP_SECURE_DN_USER has been replaced by HDFS_DATANODE_SECURE_USER. Using value of HADOOP_SECURE_DN_USER.
Starting namenodes on [Master]
Starting datanodes
Starting secondary namenodes [izbp1c010bxn27ezgr1qbZ]
Starting resource manager
Starting nodemanagers
```

If you see this, it indicates successful operation.

```
start-hbase.sh
```

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/root/hadoop-3.1.4/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/root/hbase-2.3.6/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/root/hadoop-3.1.4/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/root/hbase-2.3.6/lib/client-facing-thirdparty/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
```

If you see this, it indicates successful operation.

YCSB load:

```
cd HPCA-AE
python3 YCSB-Load.py 0
```



```
stripe_105_file_k5 10
stripe_254_file_k7 6
stripe_248_file_k8 9
stripe_198_file_k3 6
stripe_83_file_m4 9
stripe_197_file_m2 9
stripe_86_file_k4 6
stripe_92_file_k2 9
stripe_240_file_k4 2
stripe_172_file_m4 12
stripe_77_file_k5 8
stripe_10_file_k3 12
stripe_47_file_m1 16
stripe_199_file_m1 6
stripe_101_file_m4 2
stripe_88_file_k3 7
stripe_84_file_k6 2
stripe_98_file_m4 6
stripe_166_file_m1 3
stripe_37_file_k6 11
stripe_111_file_k5 14
stripe_120_file_k1 6
stripe_228_file_k1 7
stripe_108_file_k7 12
stripe_207_file_k6 4
stripe_133_file_k7 4
stripe_224_file_k4 6
stripe_181_file_k10 7
stripe_129_file_k10 4
stripe_201_file_m1 11
stripe_209_file_m2 6
```

if you see this, erasure code repair run successfully.

You can get the P99 latency of YCSB in the output:

```
[OVERALL], Throughput(ops/sec), 151.5220388805552
[TOTAL_GCS_PS_Scavenge], Count, 199
[TOTAL_GC_TIME_PS_Scavenge], Time(ms), 3784
[TOTAL_GC_TIME_%_PS_Scavenge], Time(%), 2.866796975620104
[TOTAL_GCS_PS_MarkSweep], Count, 29
[TOTAL_GC_TIME_PS_MarkSweep], Time(ms), 825
[TOTAL_GC_TIME_%_PS_MarkSweep], Time(%), 0.6250284103822901
[TOTAL_GCs], Count, 228
[TOTAL_GC_TIME], Time(ms), 4609
[TOTAL_GC_TIME_%], Time(%), 3.4918253860023936
[READ], Operations, 10099
[READ], AverageLatency(us), 418591.27319536585
[READ], MinLatency(us), 5236
[READ], MaxLatency(us), 7241727
[READ], 95thPercentileLatency(us), 3045375
[READ], 99thPercentileLatency(us), 4763647
[READ], Return=OK, 10099
[CLEANUP], Operations, 128
[CLEANUP], AverageLatency(us), 48.2421875
[CLEANUP], MinLatency(us), 1
[CLEANUP], MaxLatency(us), 4675
[CLEANUP], 95thPercentileLatency(us), 18
[CLEANUP], 99thPercentileLatency(us), 318
[UPDATE], Operations, 9901
[UPDATE], AverageLatency(us), 316486.78093121905
[UPDATE], MinLatency(us), 2302
[UPDATE], MaxLatency(us), 6500351
[UPDATE], 95thPercentileLatency(us), 2979839
[UPDATE], 99thPercentileLatency(us), 4423679
[UPDATE], Return=OK, 9901
```

You can get the repair time of erasure code in the output:

```
stripe_120_file_k1 6
stripe_228_file_k1 7
stripe_108_file_k7 12
stripe_207_file_k6 4
stripe_133_file_k7 4
stripe_224_file_k4 6
stripe_181_file_k10 7
stripe_129_file_k10 4
stripe_201_file_m1 11
stripe_209_file_m2 6
stripe_210_file_k7 3
stripe_3_file_k8 19
stripe_189_file_k9 3
stripe_176_file_k6 8
duration time = 71.906084
P25LinesFullNodeClientStream::sendFullNodeRequest() end
python3 /root/ElasticRepair/scripts/stop.py
/root/ElasticRepair/conf/config.xml
stop coordinator
OK
stop slave on 172.16.0.2
OK
stop slave on 172.16.0.3
OK
stop slave on 172.16.0.4
OK
stop slave on 172.16.0.5
OK
```

the repair throughput of erasure code is:

$64 * 200 / \text{repair time} = 64 * 200 / 71.906084 = 178 \text{ MB/s}$

Memcached-Evaluation:

Start Memcached:

```
cd ~/trace/facebook
python3 facebook-run.py 3
python3 facebook-run.py 3 (need run 2 times)
```



```

root@iZbp1c010bxn27ezgr1lqbZ:~/trace/facebook# python3 facebook-run.py 3
ssh root@Master "service memcached stop"
ssh root@Master "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave2 "service memcached stop"
ssh root@Slave2 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave3 "service memcached stop"
ssh root@Slave3 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave4 "service memcached stop"
ssh root@Slave4 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave5 "service memcached stop"
ssh root@Slave5 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave6 "service memcached stop"
ssh root@Slave6 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave7 "service memcached stop"
ssh root@Slave7 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave8 "service memcached stop"
ssh root@Slave8 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave9 "service memcached stop"
ssh root@Slave9 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave10 "service memcached stop"
ssh root@Slave10 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave11 "service memcached stop"
ssh root@Slave11 "memcached -d install -m 8192 -u root -p 11211 -c 4096 -I 1M"
ssh root@Slave12 "service memcached stop"

```

If you see this, it indicates successful operation.

Memcached Load:

```

./facebook-load
./IBM-load
./twitter-load

```

load of facebook is fast, and you will see IBM load like this:

```

root@iZbp1c010bxn27ezgr1lqbZ:~/trace/IBM# ./IBM-load
11111111111111111111111111111111
22222222222222222222222222222222

```

```

root@iZbp1c010bxn27ezgr1lqbZ:~/trace/IBM# ./IBM-load
11111111111111111111111111111111
22222222222222222222222222222222
IBM query init finished.
33333333333333333333333333333333

```

IBM load successfully.

Memcached Run:

```

python3 facebook-run.py 1
or python3 IBM-run.py 1
or twitter-run.py 1

```

![[image-20241224155312133]](C:\Users\63121\Desktop\YCSB-Evauation.assets\image-20241224155312133.pn

